HOW TO USE DOCKER

1. Install Docker on linux:

Next install needed packages:

yum install -y yum-utils device-mapper-persistent-data lvm2

Configure the docker-ce repo:   
 yum-config-manager –add-repo https://download.docker.com/linux/centos/docker-ce.repo

Finally install docker-ce:

yum install docker-ce

**How to enable docker service:**

systemctl enable docker.service

**How to start/stop/restart docker service on CentOS7/RHEL7**

systemctl start docker.service ## <-- Start docker##

systemctl stop docker.service ## <-- Stop docker##

systemctl restart docker.service ## <-- Restart docker##

systemctl status docker.service ## <-- Get status of docker##

**How to find out info about Docker network bridge and IP addresses**

ip a

ip a list docker0

2. Docker Demo:

Wordpress + Mysql:

2.1 Create image Mysql:

docker run --name demo\_mysql -e MYSQL\_ROOT\_PASSWORD=root -e MYSQL\_DATABASE=dbwordpress -e MYSQL\_USER=userwordpress -e MYSQL\_PASSWORD=password -d mysql

2.1 Create image Wordpress:

docker run --name demo\_wordpress --link demo\_mysql:mysql -p 8080:80 -d wordpress

Giải thích các tùy chọn:

--name demo\_mysql: tên của container là demo\_mysql

-e là enviroment

--link demo\_mysql:mysql: container demo-wordpress sẽ liên kết với container có tên là demo-mysql , container demo-mysql được tạo ra từ images có tên là mysql

-p 8080:80: mapping port 8080 của host

-d : container tạo ra sẽ được chạy ngầm

* wordpress: Tên của images, tên là wordpress

Test docker run trên trình duyệt: [http://ip:8080](http://ip:8080/)

3. Quản lý container và image:

list containers: docker ps -a hoặc docker ps -aq

list the last container created: docker ps -l

stop container: docker stop id container

start container: docker start id containers

remove container: docker rm id container

remove all container: docker rm $(docker ps -aq)

remove image: docker rmi tên image

remove all images: docker rmi

Goto container cli: docker exec -i -t id container /bin/bash

Check IP container: docker inspect --format '{{ .NetworkSettings.IPAddress }}' id containers

Check all IP container: docker inspect --format '{{ .NetworkSettings.IPAddress }}' $(docker ps -q)